REMARKS

This Amendment is submitted in response to the December 8, 2005 Final Office Action issued in connection with the above-identified patent application. Independent claims 1, 22 and 40 have been amended as indicated above. No new matter has been added. Upon entry of this Amendment, the pending claims will be amended independent claim 1, with claims 2-5 and 7-21 depending therefrom, amended independent claim 22, with claims 23 and 26-39 depending therefrom, and amended independent claim 40, with claims 41-43 depending therefrom. It is respectfully requested that the Examiner review and consider the amended claims in view of the following remarks.

Claim 1 is directed to a method for executing a target application on a computer. The claim has been amended to recite that the configuration file which is searched "is external to the target application". Claim 22 which is directed to a system including a supervisor software module has been amended to state that the supervisor software module is "external to the target application". Lastly, claim 40 which is directed to a system having a supervisor software module has been amended to state that the supervisor software module is "external to the application software module". Support for these amendments can be found on page 7, paragraph 28 of the specification which describes the Java Application Supervisor (JAS) as being "an external supervisor for Java Applications".

In the Office Action, the Examiner has maintained a rejection of the claims as allegedly rendered obvious from the combination of the article titled "Enhancing Java Server Availability With JAS" (hereinafter "Reinhard") in combination with U.S. Patent No. 6,757,897 (Shi et al.). Applicant respectfully traverses this rejection.

Dealing first with the Reinhard reference, the Office Action states that this reference "does not teach [that] said configuration file specifies periodic checking for a thread starvation condition." (See P. 3, lines 3-4 of the December 8, 2005 Office Action). This deficiency is allegedly taught by Shi et al, which discloses the scheduling of a task to prevent task or process starvation conditions. However, Shi et al. operates in an entirely different manner as compared to the now-amended independent claims and, in fact, teaches away from the invention as now claimed.

A significant difference between Shi et al. and the present invention, as now claimed, is that Shi et al. is directed to a task internal system. Such a system requires the task itself to trigger a thread starvation condition if the task takes too long to be performed so that the processor (CPU) can perform other tasks and then return to the "starved" task.

A task internal system as is disclosed in Shi et al. has at least two primary drawbacks. First, such a system inherently requires a task to commence. If the task takes too long to complete, a yield signal is then generated by the task to free the processor so that it can work on other tasks and then return to the starved task. According to Shi et al., "allowing a task to yield time to one or more other tasks is advantageous over conventional systems since the decision to yield processor time is given to the task itself and thus task starvation can be better controlled". (Shi et al., col. 4, lines 25-29) (emphasis added). In col. 4, lines 60 through col. 5, line 5 – which has been relied upon in the Office Action in rejecting the claims – Shi et al. explain that "upon the start of performance of the first task" an elapsed performance time is monitored and "a step of generating the yield signal from within the first task" is performed. The problem, however, is that if a task never commences in the Shi et al. system, there will be no time interval to measure and, therefore, no way for the task to generate a yield signal to free up the processor to perform other tasks. This scenario is commonly referred to in the art as a "deadlock condition", which is a subset of a thread starvation condition.

A second drawback of the Shi et al. system is that each task that is intended to have yield capabilities must include instructions, i.e. software coding, etc. to perform this functionality.

In contrast, the invention as is now claimed does not rely on the tasks themselves to trigger a thread starvation condition. Rather, and with reference to amended claim 1, for example, a configuration file "that is external to the target application" specifies periodic checking for a thread starvation condition. Because the configuration file is external to the target application, it does not require the target application to commence in order to detect a thread starvation condition and, in that event, yield processor power to another application. In other words, the present invention will also prevent deadlock conditions from occurring. Moreover, the invention of the subject application is a task transparent system and method because the tasks themselves do not need to be modified to include thread starvation detection coding or instructions. In other words, an external entity, i.e. a configuration file, is used to monitor thread starvation conditions for all tasks. This allows target applications to be easily added to a system without embedding or otherwise including instructions to trigger a yield signal.

For at least the reason that the system and method of Shi et al. is directed to an internal thread starvation detection system wherein the detection is internal to the performed tasks, the combination of Shi et al. with Reinhard does not render the now-amended claims unpatentble as being obvious. Accordingly, it is believed that all pending claims are now in condition for immediate allowance.

It is believed that no fees or charges are required at this time in connection with the present application. However, if any fees or charges are required at this time, they may be charged to our Patent and Trademark Office Deposit Account No. 03-2412.

Respectfully submitted,

COHEN, PONTANI, LIEBERMAN & PAVANE

By _____

Edward M. Weisz
Reg. No. 37,257
551 Fifth Avenue, Suite 1210
New York, New York 10176
(212) 687-2770

Dated: March 7, 2006